Engineering Applications of Artificial Intelligence 37 (2015) 54–70

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

DFP-SEPSF: A dynamic frequent pattern tree to mine strong emerging patterns in streamwise features

Fatemeh Alavi, Sattar Hashemi *

CSE and IT Dept., Engineering Campus Number 2, Mollasadra Ave., P.O. Box: 71348 51154, Shiraz, Iran

ARTICLE INFO

Article history:
Received 18 April 2014
Received in revised form
27 June 2014
Accepted 15 August 2014

Keywords:
Strong emerging patterns
Feature streams
Dynamic frequent pattern tree

ABSTRACT

Mining a minimal set of strongly predictive emerging patterns from a high dimensional dataset is a challenging issue for making an accurate emerging pattern classifier. The problem becomes even more severe when features are not available as a whole; in this scheme, features are emerged one by one instead of having all features at hand before learning process gets started. In this study, we propose a novel dynamic structure to construct the frequent pattern tree on arrival of new features and to mine emerging patterns online. DFP-SEPSF, a *D*ynamic *F*requent *P*attern tree to mine *S*trong *E*merging *P*atterns in *S*treamwise *F*eatures, offers an efficient bottom up approach to construct an Unordered Dynamic Frequent Pattern tree (UDFP-tree) and an Ordered Dynamic Frequent Pattern tree (ODFP-tree). Moreover, the proposed framework extracts Strong Emerging Patterns (SEPs) to build an accurate and fast classifier that can deal with noise. Our experimental evaluations indicate the effectiveness of the proposed approach in comparison with other state-of-the-art methods, in terms of predictive accuracy, emerging pattern numbers, and running time.

1. Introduction

Discovering patterns which represent discriminating knowledge between different classes, is an important issue in data mining. The concept of *emerging pattern* (Dong and Li, 1999) has been proposed to capture sharp changes between different classes. An Emerging Pattern (EP in short) is a multi-variate feature whose support value increases significantly from one class to another (Dong and Li, 1999; Zhang et al., 2000a). A Jumping Emerging Pattern (JEP) is a type of EPs whose support changes abruptly from zero in one class to non-zero in another class; where the ratio of support-increase is infinite (Li et al., 2001).

Emerging pattern research mainly utilizes mined patterns for classification purposes, even for imbalanced datasets (Dong et al., 1999; Li et al., 2000a; Fan and Kotagiri, 2002; García-Borroto et al., 2011). EPs are used in many real-world applications, such as failure detection (Lo et al., 2009), customer behavior detection (Song et al., 2001), disease diagnosis (Li et al., 2003), and discovering knowledge in gene expression data (Li and Wong, 2002; Boulesteix et al., 2003; Mao and Dong, 2005; Fang et al., 2012). The number of EPs is exponentially increasing in massive datasets which include millions or billions of features, such as image processing datasets, gene expression data, text data, and so on. The key idea in mining emerging patterns is to extract a small set of the strongly predictive emerging patterns to improve the predictive accuracy.

Traditional learning systems ignore considerable computational cost involved in generating features. They assume that all the features associated with the training data are readily available at the start of the learning process (Perkins and Theiler, 2003). Feature generation algorithms, such as pairwise interactions (Foster and Stine, 2004) and Statistical Relational Learning (SRL) (Dzeroski and Lavrac, 2001; Džeroski, 2003) methods, mostly generate tens or hundreds of thousands features. The number of generated features is restricted by the amount of CPU time available to run queries and memory constraints (Zhou et al., 2006).

Accordingly, in feature generation methods, we usually encounter three challenging research issues: (1) can we spend a long time waiting for all generated features and then adopt existing algorithms (Yu et al., 2010)? (2) Generating features are expensive. Generating 100,000 features can easily take 24 CPU hours, while millions of features may be irrelevant (Zhou et al., 2006). (3) Because of memory constraints, millions or billions of generated features do not fit easily into memory.

To solve these problems, the notion of *feature stream* has been presented to conduct the feature generation process in an uncertain feature space over time (Ungar et al., 2005; Zhou et al., 2006). In this scheme, the number of the training instances is fixed, but the feature dimension increases over time, in other words, feature

* Corresponding author. Tel.: +98 711 613 3544; fax: +98 711 647 4605.
E-mail addresses: alavi@cse.shirazu.ac.ir (F. Alavi), s_hashemi@shirazu.ac.ir (S. Hashemi).

http://dx.doi.org/10.1016/j.engappai.2014.08.010
0952-1976/